APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: MANAGING AND CONTROLLING USER APPLICATIONS WITH NETWORK SWITCHES

APPLICANT: Christopher H. Claudatos and Magnus B. Hansen

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 321 388 646 US

August 27, 2003
Date of Deposit

MANAGING AND CONTROLLING USER APPLICATIONS WITH NETWORK SWITCHES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority based on United States Provisional Application No. 60/406,713 for "Managing and Controlling User Applications with Network Switches", filed August 28, 2002, the disclosure of which is incorporated here by reference in its entirety.

BACKGROUND

This invention relates to network switching, and more particularly to Layer 2 through Layer 7 switching.

The OSI (Open System Interconnection) Model is an ISO standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the applications layer in one station, and proceeding to the physical layer and back up the hierarchy.

The layers are defined as:

Applications Layer 7 provides interface to end-user processes and standardized services to applications.

Presentation Layer 6 specifies architecture-independent data transfer format, encodes and decodes data, encrypts and decrypts data, compresses data.

Session Layer 5 manages user sessions and reports upper-layer errors.

Transport Layer 4 manages network layer connections and provides reliable packet delivery mechanism.

Network Layer 3 addresses and routes packets.

Data Link Layer 2 frames packets and controls physical layer data flow.

Physical Layer 1 interfaces between network medium and network devices. Also defines electrical and mechanical characteristics.

SUMMARY OF THE INVENTION

In general, in one aspect, the invention provides method and apparatus, including computer program products, for processing data packets in a computer network, the data packets including information from one or more of Layers 2 through 7 of the OSI model. The method includes configuring a multilayer switch to process data packets at wire speed based on one or more user defined packet policies, each user defined packet policies specifying information for one or more of Layers 4 through 7, receiving a data packet at the multilayer switch, the data packet including information from one or more of Layers 2 through 7 of the OSI model. The method determines if there is a match between the data packet and one or more of the packet policies, each packet policy authorizing matching data packets to use the computer network. If there is a matching packet policy authorizing the data packet, the data packet is routed using a Layer 2-3 switch. If there is no matching packet policy authorizing the data packet, the data packet is blocked.

Implementations of the invention include one or more of the following features. The user defined packet policies can include timed packet policies, where the timed packet policies are active during specified date or time intervals. Determining if there is at least one matching packet policy can include determining if there is a currently active matching timed packet policy. The user defined packet policies can authorize data packets being transmitted or received by authorized users, applications, physical ports, application ports, IP addressess, or MAC addresses. Blocking the data packet can include discarding the data packet, logging the data packet, or forwarding the data packet to a multilayer switch application for processing.

In another aspect, the invention is directed to a method for configuring a multilayer switch to process data packets at wire-speed based on one or more user defined packet policies, each user defined packet policy specifying information for one or more of Layers 2 through 7. Data packets are received at the multilayer switch, the data packets including information from one or more of Layers 2 through 7 of the OSI model. The method determines if there is a match between the data packet and one or more packet policies that specify a second packet policy to be applied to the matching data pockets. The second packet policy can have one or more policy action fields. If there is a matching packet policy specifying a second packet policy, the data packet is processed based on the policy action fields of the second packet policy.

Implementations of the invention include one or more of the following features. The matching packet policy can specify the application of a preexisting second packet policy, and processing the data packet can include defining the preexisting second packet policy and processing the data packet based on the policy action fields of the preexisting second policy. The matching packet policy can specify the application of a dynamically created second packet policy. Processing the data packet can include creating the second packet policy and processing the data packet based on the policy action fields of the created second packet policy. Processing the data packet can include routing the data packet using the Layer 2-3 switch.

In another aspect, the invention is directed to a method for configuring a multilayer switch to process data packets at wire-speed based on one or more user defined packet policies. Each user defined packet policy specifies information for one or more of Layers 4 through 7. A data packet is received at the multilayer switch and the data packet includes information from one or more of Layers 2 through 7 of the OSI model. The method determines if there is a match between the data packet and one or more packet policies that assign a quality of service (QoS) metric to matching data packets. If there is a matching packet policy assigning a QoS metric to the data packet, a priority is determined for the data packet based on the assigned QoS metric. The data packet is routed using a Layer 2-3 switch based on the priority.

Implementations of the invention include one or more of the following features. The QoS metric can specify prioritization, bandwidth allocation, minimum bandwidth allocation, maximum allocation, or network access permission for the data packet. Assigning a QoS metric can include assigning a QoS metric based on application, application type, application port, physical port, elapsed time, time of day, day of week, date, or time intervals. Assigning a QoS metric can include assigning a QoS metric for individual users, work groups, VLAN, subnets, networks, IP addresses, IP address range, MAC addresses, and MAC address range.

In another aspect, the invention is directed to a method for configuring a multilayer switch to process data packets at wire-speed based on one or more user defined packet policies. Each user defined packet policy specifies information for one or more of Layers 4 through 7. A data packet is received at the multilayer switch, the data packet including information from one or more of Layers 2 through 7 of the OSI model. The data packet is part of a network flow representing access to a specific website. The method determines if there is a match between the data packet and one or more of the packet policies, where the matching packet policies authorize access to the specific website. If there is a matching packet policy authorizing access to the specific website, the data packet is routed using the Layer 2-3 switch. If there is no matching packet policy authorizing access to the specific website, the data packet is blocked.

Implementations of the invention can include one or more of the following features. The user defined packet policies can include timed packet policies where the timed packet policies are active during specified date or time intervals. Determining if there is at least one matching packet policy can include determining if there is a currently active timed matching policy authorizing access to the specific website. The user defined packet policies can authorize access to specific websites by authorized users, applications, physical ports, application ports, IP addresses, or MAC addresses. Blocking the data packet can include discarding the data packet, logging the data packet, or forwarding the data packet to a multilayer switch application for processing.

In another aspect, the invention is directed to a method for configuring a multilayer switch to process data packets at wire-speed based on one or more user defined packet policies, each user defined packet policy specifying information for one or more of Layers 4 through 7. A data packet is received at a particular port of the multilayer switch. The data packet includes information from one or more of Layers 2 through 7 of the OSI model. The method determines if there is a match between the data packet and one or more of the packet policies, where each packet policy blocks matching data packets received at the particular port from utilizing the computer network. If there is a matching packet policy blocking the data packet, the data packet is blocked. If there is no matching packet policy blocking the data packet, the data packet is processed.

Implementations of the invention can include one or more of the following features. The user defined packet policies can block data packets received at the particular port for data packets having a subnet address, a range of subnet addresses, a host address, or a range of host addresses.

In another aspect, the invention is directed to a method for configuring a multilayer switch to process data packets at wire-speed based on one or more user defined packet policies, each user defined packet policy specifying information for one or more of Layers 4 through 7. A data packet is received at the multilayer switch. The data packet includes information from one or more of Layers 2 through 7 of the OSI model. The method determines if there is a match between the data packet and one or more of the packet policies, where each packet policy specifies that surveillance is to be performed on the data packet. If there is a matching packet policy specifying surveillance, the data packet is mirrored to a specified location and the data packet is processed using the multilayer switch. Implementations of the invention include routing the data packet using a Layer 2-3 switch.

The invention can be implemented to realize one or more of the following potential advantages. User defined quality of service metrics can be used to provide bandwidth control and prioritization of packets for wired and wireless networks. By permitting only data packets authorized by a user defined packet policy to utilize the computer network, traffic from unauthorized applications is reduced or eliminated so that more of or the entire network bandwidth is available for authorized applications. The user defined packet policies can be used to control Web usage for all users of the computer network or for specific users. The user defined packet policies can be used to grant access only to specific websites or to deny access to specific websites. Unauthorized access using IP spoofing can be prevented by blocking access to the computer network for all unauthorized traffic entering on a particular port of the multilayer switch. Creating new packet policies based on historical network usage patterns allows a dynamic response to actual network usage. The network administrator can use Layer 2-7 information to identify data packets to be cloned and use the cloned packets for surveillance. One implementation of the invention can provide all of the above advantages.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Further features, aspects, and advantages of the invention will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a network topology including a multilayer switch.

FIG. 2A is a block diagram of an exemplary implementation of the switch.

FIG. 2B is a block diagram illustrating an alternative switch implementation including a time triggered action unit (TTA).

FIG. 2C is a block diagram of an implementation of the switch including a central management unit (CMU).

FIG. 3 is a block diagram illustrating the components of a packet policy.

FIG. 4 is a block diagram illustrating the types of packet policies that may be requested by the user.

FIG. 5 is a block diagram illustrating a method of operation of the packet filter engine.

FIG. 6 is a block diagram illustrating the components of a timed policy request to be processed by the TTA.

FIG. 7 is a flow diagram illustrating a method of processing a timed policy request.

FIG. 8 is a flow diagram illustrating activation of a packet policy scheduled using a timed policy request.

FIG. 9 is a flow diagram illustrating a method for processing data packets where the processing is prioritized based on a user-defined quality of service (QoS) metric.

FIG. 10 is a flow diagram illustrating the processing of data packets in a computer network where only packets authorized by an active packet policy are routed using the computer network.

FIG. 11 is a flow diagram illustrating a method for limiting access to websites.

FIG. 12 is a flow diagram illustrating a method for blocking access to the computer network for all unauthorized traffic entering on a particular port of the multilayer switch.

FIG. 13 is a flow diagram illustrating a method for dynamically applying additional packet policies based on initial criteria set within a first packet policy.

FIG. 14 is a flow diagram illustrating a method for performing surveillance on network traffic.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

FIG. 1 shows a network topology including a local area network (LAN) 100, including a server 102, several workstations (W/S) 104, a firewall 106, and multilayer switch 108. The LAN 100 is connected to an external network, e.g., the Internet 114, through the firewall 106. The LAN 100 is also connected to a second LAN 116 through the firewall 106. The second LAN 116 includes a web server 110, an email server 112, a server 102, several workstations 104, a firewall 106 and one or more multilayer switches 108. The computers, servers and other devices in the LAN are interconnected using a number of data transmission media such as wire, fiber optics, and radio waves. Each router 118 processes packets based on Layer 3 information and routes the packets through the network. The multilayer switch 108 processes and routes packets at Layer 2 and Layer 3, but modifies the routing behavior based on the processing of information contained in Layers 2 through 7 of the packet. The multilayer switch 108 processes the information in Layer 2 through 7 of the packet in an amount of time available for routing a packet at Layer 2 (wire-speed).

FIG. 2A shows a block diagram of an exemplary implementation of the switch 108. The switch 108 implements one or more packet policies that specify the action to be performed by the switch 108 when a packet is received that matches the conditions set in the policy. The switch 108 includes a packet policy manager (PPM) 210 and a packet filter engine (PFE) 230. The user or network administrator 225 interacts with the PPM 210 through the user interface 220 to specify the requested packet policies to be implemented by the switch 108. In one implementation, the switch 108 includes an HTTP server and the user interface displays a web page that can be used by the user 225 to specify the requested packet policies. The PPM stores the requested packet policies in the packet policy repository (PPR) 205. In one implementation, the PPM 210 assigns a packet policy identifier for each requested packet policy and the packet policies can be retrieved from the PPR 205 using the packet policy identifier. The PPM 210 transmits the requested packet policies to the PFE 230 in order to activate the packet policies. The PFE 230 stores the active packet policies along with the packet policy identifier for each active policy. The switch 108 receives data packets using the incoming packet interface 240. A data packet includes data being communicated in a computer network that has been packetized. A data packet also includes TCP/IP packets. The PFE 230 screens incoming data packets to determine if they match one of the requested packet policies. If the received data packet matches one of the requested packet policies, the PFE 230 can block the received data packet or modify the data packet as requested by the matching packet policy before routing. If the received data packet is not blocked by the PFE 230, it is routed by the Layer 2-3 switch 235 using the out going packet interface 245.

FIG. 3 is a block diagram illustrating the components of a packet policy 300. Each packet policy 300 can have an associated packet policy identifier 305 that can be used to access the packet policy. The packet policy 300 contains a policy byte pattern 310 and one or more policy action fields 315. Each policy action field 315 can also have an associated policy action value 320. The policy action field 315 specifies the processing of the received packet including whether the received packet should be routed, blocked, redirected, or cloned. The policy action field 315 can also specify modifications to be performed on the packet before it is routed. An incoming packet matches the packet policy 300 if the incoming pattern contains a sequence of bytes identical to the policy byte pattern 310. The policy action fields 315 specify one or more actions to be performed when a matching packet is received. The policy action value 320 specifies additional optional parameters for the policy action field 315. Table I is an exemplary list of values for the policy action field 315 along with a description of the action to performed for each value.

TABLE I

Action	Function	Action Value
None	No sub service is selected in this policy.	None
Discard	Drops packets that match this policy	None
Flow Meter	Regulates the percentage of 1-100 bandwidth for packets that match this policy. The percentage is specified in the policy action value.	10/100 ports: 1=1Mbps Gigabit ports: 1=8Mbps Example: 5
Mirror to Port	Mirrors packets that match this policy to the mirrored to port. Port mirroring must be enabled on the switch. The mirror port is specified when the switch is configured.	None
Redirect	Changes port of Egress for packets that match this policy. The Egress port is specified in the policy action value.	Ports 1- 26, Example: 24
Prioritize	Internally prioritizes packets that match this policy. The policy action value specifies the priority.	0-7 Example: 5
Do Not Drop	If a policy is created to drop a certain type of traffic this option can be selected to not discard packets that match this policy.	None
Change 802.1p Tag	Redirects packet to a new CoS queue as specified by the policy action value.	0-7 Example: 3
Change IPTOS	Redirects packet to a new CoS queue as specified by the policy action value.	0-7 Example: 3
Change IPTOS to 802.1p	Matches IPTOS to 802.1p	None
IP DiffServ	Modify the IP header to insert the "differential services code point" (DSCP) as specified by the policy action value.	0-31 Example: 11

FIG. 4 is a block diagram illustrating the types of packet policies 400 that may be requested by the user. The requested packet policies can be selected from predefined packet policies 405 or expert packet policies 410. Referring to FIG. 3, expert packet policies 410 are user defined packet policies for which the user provides the policy byte pattern 310, the policy action fields 315, and the associated policy action values 320. Predefined packet policies 405 consist of packet policies that are used by a large number of users. The PPM (210, FIG. 2) provides the policy byte pattern 310 for predefined packet policies 405 and the user is not required to provide a byte pattern for these policies. The PPM 210 also provides default policy action fields 315 and policy action values 320 for each predefined packet policy 405. In one implementation, the user can customize a predefined packet policy 405 by modifying the policy action fields 315 and policy action values 320. Predefined packet policies 405 can include packet policies for commonly used applications like Yahoo Messenger, Microsoft Netmeeting, or interactive networked computer games. Predefined packet policies 405 can also include packet policies for known network security attacks like IP spoofing, and to block access to specific URLs.

FIG. 5 is a flow diagram illustrating the method of operation of the PFE (230, FIG. 2). Incoming packets are received (step 500), and analyzed in the PFE 230 using the active packet policies (step 505). If there is no matching packet policy ("no" branch of decision step 510), the packet is routed by the Layer 2-3 switch (235, FIG. 2) (step 515). If there is a matching packet policy ("yes" branch of decision step 510), the actions specified in the policy action fields (315, FIG. 3) are performed (step 520). If the packet is not blocked by the policy action fields 315 of the matching policy ("no" branch of decision step 525), it is routed by the Layer 2-3 switch 235 (step 515). If the packet is blocked by the policy action fields 315 of the matching policy ("yes" branch of decision step 525), the blocked packet is forwarded to the multiplexer (250, FIG. 2) along with the packet policy identifier (305, FIG. 3) of the matching packet policy (step 530).

Referring to FIG. 2A, the multiplexer 250 forwards the blocked packet and the blocked policy identifier to one or more switch applications 255 running on the switch. In one implementation, the blocked packet and the associated packet policy identifier are also sent to other network devices external to the switch 108 for further processing. Switch applications 255 and external network devices can avoid analyzing the blocked packet by using the associated packet policy identifier to identify the matching policy for the blocked packet. In one exemplary embodiment of the switch 108, one of the network applications 255 can be a network address translation (NAT) application that receives and processes blocked NAT packets. In another exemplary embodiment of the switch 108, one of the network applications 255 can be a network security application that analyzes blocked packets for known attack signatures to determine if an attempted network security intrusion is in progress. The network security application can also transmit additional packet policies to the PFE 230 through the PPM 210 to block an attempted network security intrusion.

FIG. 2B is a block diagram illustrating an alternative implementation of the switch 108 including a time triggered action unit (TTA) 215. The TTA 215 allows the user to schedule timed packet policies that are used to filter incoming packets only during the specified time periods. The TTA 215 schedules the timed packet policies using a time reference obtained from a real time clock 265. The user can specify that a requested packet policy is to be used only during specified time periods. In one implementation of the switch 108, the TTA 215 is also used to schedule switch applications 255 to run during certain specified time periods.

FIG. 2C is a block diagram illustrating another implementation of the switch 108 including a central management unit (CMU) 270. As described later, the CMU 270 is used for performing firmware and configuration updates.

FIG. 6 is a block diagram illustrating a timed policy request 600 to be processed using the TTA (215, FIG. 2). The timed policy request 600 includes a packet policy identifier 605, and one or more pairs of start time 610 and end time 615 values. The packet policy identifier 605 identifies a policy that already been programmed by the user. The start time 610 and the end time 615 indicate the activation time and de-activation time for the policy identified by the packet policy identifier 605. If there is no end time for timed policy request 600, the policy identified by the packet policy identifier 605 is never deactivated after activation. A timed policy request 600 with no start time is used to de-activate an active policy identified by the packet policy identifier 605 at the specified end time 615. In one implementation, the timed policy request includes the packet policy to be scheduled instead of the packet policy identifier 605.

FIG. 7 is a flow diagram illustrating a method of processing a timed policy request (400, FIG. 4). Referring to FIG. 2 and FIG. 4, the PPM 210 receives a timed policy request 400 (step 700). The PPM 210 validates the timed policy request 400 by verifying that the packet policy identifier 605 identifies a packet policy that exists in the PPR 205 (step 705). If the timed policy request is invalid, an error is returned to the user (step 710). If the timed policy request is valid, the timed policy request is forwarded to the TTA 215 to be scheduled (step 715). The TTA 215 schedules a triggering event for each start time 610 and end time 615 included in the timed policy request 600 (step 720).

FIG. 8 is a flow diagram illustrating activation of a packet policy scheduled using a timed policy request (400, FIG. 4). Referring to FIG. 2 and FIG. 4, the TTA 215 receives a policy triggering event (step 800), and forwards the policy triggering event to the PPM 210 along with the packet policy identifier 605 associated with the triggering event (step 505). The PPM 210 retrieves the packet policy associated with the triggering event from the PPR 205 using the packet policy identifier 605 (step 810). If the received triggering event is associated with a start time 410 ("yes" branch of decision step 815), the PPM 210 transmits the retrieved policy to the PFE 230 for activation (step 820). If the received triggering event is associated with an end time 615 ("no" branch of decision step 815), the PPM transmits the retrieved packet policy to the PFE 230 for de-activation (step 825).

Techniques for implementing a switch, such as the switch 108, are described in U.S. Application No. 10/445,293, titled "Switch for Local Area Network," to Sean Hou, William R. Ge, Daniel Yin Yung Ching, Keith M. Andrews, Christopher H. Claudatos, and Magnus B. Hansen , filed on May 22, 2003, which is incorporated by reference herein.

FIG. 9 is a flow diagram illustrating a method for processing data packets where the processing is prioritized based on a user-defined quality of service (QoS) metric. The data packet is received at a multilayer switch 108 (step 900) and analyzed in the packet analysis engine 230 using active packet policies (step 905). The packet analysis engine determines if there is a matching packet policy that specifies a QoS metric for the received packet (step 910). If there is no matching packet policy specifying a QoS metric set for the received packet ("no" branch of decision step 910), a default QoS metric is assigned to the received packet (step 930). If there is a matching packet policy specifying a QoS metric for the received packet ("yes" branch of decision step 910) the packet is processed according to the specified QoS metric (step 915). If the specified QoS metric for the received packet denies network access permission to the received packet ("yes" branch of decision step 920), the received packet is discarded (step 935). If the specified QoS metric does not deny network access for the received packet ("no" branch of decision step 920), the QoS metric specified by the matching packet policy is assigned to the received packet (step 940). The multilayer switch determines a priority for the received packet based on the assigned QoS metric and routes the received packet according to the priority (step 945).

The assigned QoS metric can specify a priority for processing the received packet. In one implementation, the a priority is assigned to each received packet, the priority values ranging from level 1 through level 7. Received packets having a higher priority level are processed faster and/or allocated a greater portion of available network bandwidth than received packets having a lower priority level, e.g., packets having level 1 priority are processed faster than packets having level 2 priority. The assigned QoS metric can also specify bandwidth allocation or network access permission for the received packet. In one implementation, the assigned QoS metric specifies a maximum bandwidth and a minimum bandwidth that is allocated to the received packet. The received packet can be classified into a number of QoS classes and a QoS metric can be assigned for each QoS class. The received packet can be assigned a QoS class based on the application generating the packet, the type of application generated in the packet, the application port, the physical port, the elapsed time, time of day, day of week, date, or time entry. The received packet can also be assigned a QoS class based on the identity of the user generating the packet, the work group or the user generating the packet, VLAN subnet address, network address, source or destination IP address, or MAC ID range. QoS metrics can be specified for both wired and wireless networks.

FIG. 10 is a flow diagram illustrating the processing of data packets in a computer network where only packets authorized by an active packet policy are routed using the computer network. A data packet is received at a multilayer switch 108 (step 900) and the received packet is analyzed in the packet analysis engine 230 using active packet policies (step 905). If there is a matching packet policy authorizing the received packet ("yes" branch of decision step 1010), the received packet is routed using the computer network (step 1015). If there is no matching packet policy authorizing the received packet ("no" branch of decision step 1010) the received packet is forwarded for further processing (step 1020).

Further processing of the received packet in step 1020 can include blocking, discarding, forwarding, mirroring to a predefined physical port of the multilayer switch, logging, and forwarding the received packet.

Packet policies authorizing received packets in step 1010 can be used to authorize packets being transported or received by authorized users, applications, physical ports, application ports, addresses, source or destination IP addresses, and MAC addresses. Packet policies authorizing the processing of received packet can be timed packet policies that are active only during specified times of the day or during specified date and time intervals. The packet policies can also be used to only grant authorized users access to specific applications, websites, or locations on the network. A received packet that is blocked by step 1020 can be forwarded to another switch application for further processing.

The method in FIG. 10 can be used to eliminate traffic from unauthorized applications, allowing the network administrator to make network bandwidth available for authorized applications. As a result of allowing only authorized traffic based on user defined packet policies, the network bandwidth is not consumed by unauthorized traffic resulting from unauthorized applications installed by the users of the computer network. Blocking all the unauthorized traffic also prevents computer viruses, worms, trojans, etc., from using the network to spread themselves or launch attacks. Blocking all unauthorized traffic also prevents denial of service attacks to be launched using the completed network. In addition, any unauthorized packet, i.e., a packet that is not authorized by an active packet policy can be sent to a predefined port for the purpose of reporting. Data regarding blocked packets can be used to identify any workstations that are infected with a virus, failing technically, or being used to breach the security of the computer network. The method also makes it possible to restrict a user's access to specific applications and locations on the network. In one implementation, the user or a group of users can be blocked from general access to the Internet but can be given access to specific websites. For example, a corporation can block its warehouse employees from accessing the Internet and only allow them to access specific websites required for their work, e.g., FedEx, UPS, DHL, and the U.S. Post Office.

FIG. 11 is a flow diagram illustrating a method for limiting access to websites. Packet policies are defined that implement corporation wide or user specific policies to control website access. In addition, packet policies can also be defined to limit access to websites based on a specific IP address, a specific MAC address, or during a specified time interval. The user or network administrator defines policies that permit access only to specific authorized websites. A data packet is received at a multilayer switch (step 900) and analyzed using the packet analysis engine using active packet policies (step 905). If the packet analysis engine determines that the data packet is part of a website access, it applies packet policies defined to limit website access. If there is a matching packet policy authorizing access to the web site ("yes" branch of decision step 1110), the received packet is processed (step 1120). If there is no matching packet policy authorizing access to the website ("no" branch of decision step 1110), the data packet is blocked (step 1115). The blocked packet can be forwarded to another switch application or it can be forwarded to a specific port or switch application for logging or reporting purposes.

FIG. 12 is a flow diagram illustrating a method for blocking access to the computer network for all unauthorized traffic entering on a particular port of the multilayer switch. A data packet is received at a multilayer switch 108 (step 900) and analyzed using the packet analysis engine using active packet policies (step 905). Port specific packet policies can be used to specify the type of data packets that are blocked from accessing the computer network using the specified port. If there is a matching packet policy blocking access to the computer network for the data packet received on a particular port ("no" branch of decision step 1210), the data packet is blocked from accessing the computer network (step 1215). The blocked packet can be forwarded to another port or switch application for logging or reporting purposes. If there is no matching packet policy blocking access to the computer network for the received data packet ("yes" branch of decision step 1210), the received packet is processed at the multilayer switch (step 1220). The method can be used to prevent IP spoofing where an attacker gains unauthorized access to computers and networks by sending messages to computers with an IP address indicating that a message is coming from a trusted host. IP spoofing can be prevented by selecting the subnet or host address that is blocked from accessing the network on a particular port of the multilayer switch.

FIG. 13 is a flow diagram illustrating a method for dynamically applying additional packet policies based on initial criteria set within a first packet policy. A data packet is received at the multilayer switch (step 900) and the received packet is analyzed at the packet analysis engine using active packet policies (step 905). If there is a matching packet policy specifying the creation of a new packet policy ("yes" branch of decision step 1310), the new packet policy is created (step 1315) and the created packet policy is applied to the received data packet (step 1320). This method can be used to dynamically create new packet policies based on initial criteria set within the first policy. This allows greater control over the traffic and gives the switch ability to provide dynamic responses to the network traffic. If the matching packet policy specifies the application of a second preexisting packet policy ("no" branch of decision step 1310), the received packet is processed in accordance with the specified preexisting packet policy. The method can be used to dynamically construct new packet policies based on historical application and bandwidth usage data. For example, if historical network usage data indicates a higher bandwidth usage by the finance department at the end of each month, a packet policy granting a higher priority or a higher bandwidth to the finance department at the end of each month can automatically be created at the multilayer switch.

FIG. 14 is a flow diagram illustrating a method for performing surveillance on network traffic. A data packet is received at the multilayer switch (step 900) and the received packet is analyzed at the packet analysis engine using active packet policies (step 905). If there is a matching packet policy specifying that surveillance is to be performed ("yes" branch of decision step 1410), the received data packet is mirrored to a desired location at wire-speed with no delay (step 1425). The received data packet is forwarded to its original destination at the same time (step 1430). Surveillance based on IP addresses and/or MAC addresses can be used to monitor the network traffic of individual users. Additional information from the data packet and the packet header can be used to perform surveillance of individual users based on the content of the network traffic. Port surveillance can be performed by specifying a source port, destination port, and data direction that is to be monitored. Flow surveillance can be performed by specifying the source and destination IP address, MAC address, subnet address, or UDP/TCP port that is to be monitored.

Processing the data packet includes routing the data packet using a multi-layer switch. Processing the packet also includes allocating bandwidth, specifying a minimum bandwidth, and specifying a maximum bandwidth for the data packet. Processing the packet can also include redirecting the packet to another port of the network device 215 processing the data packet, redirecting the data packet to another device connected to the network 220, mirroring the packet to a particular physical port of the network device 215, prioritizing the data packet, and counting discarded data packets. The network device 215 receiving the data packet can also modify the network rights descriptor for the data packet, or add a secondary network rights descriptor to the data packet. The secondary or modified network rights descriptor is used in the same manner as the original network rights descriptor. In one implementation, the network device 215 is a multi-layer switch that processes the data packet according to user-defined packet policies for the network rights descriptor contained in the data packet.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Method steps of the invention can be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method steps can also be performed by, and apparatus of the invention can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the steps of the invention can be performed in a different order and still achieve desirable results.

What is claimed is: